REMARKS

Claims 1-22, 25-34, and 37-40 have been examined, will all claims remaining rejected based on prior art. Claims 1-5, 9-16, 20-22, 25, 26, 29-32, 34, 35, and 37-40 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Hobbs et al. (U.S. Patent No. 5,197,138; hereinafter "Hobbs") in view of Wilson et al. (U.S. Patent No. 6,944,736; hereinafter "Wilson"). Claims 6-8, 17-19, 27, 28, 33, and 36 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Hobbs in view of Wilson and Radhakrishna (U.S. Patent No. 6,823,414).

Independent claim 1 recites "assigning a unique interrupt priority value to each of a plurality of interrupts; ... and processing, by a thread execution logic, a requested interrupt only when the unique interrupt priority value assigned to the requested interrupt is higher than the global interrupt threshold value."

The Examiner asserts that Hobbs teaches, in column 2, lines 54-57, processing a requested interrupt only when the interrupt priority value of the requested interrupt is higher than the interrupt threshold value. However, what Hobbs actually teaches is that an interrupt will not be recognized or serviced by the processor until the priority of the code thread is lower than the priority of the interrupt. In other words, Hobbs compares the priority of the interrupt to the priority of the thread. This is in contrast to independent claim 1, which requires processing a requested interrupt only when the unique interrupt priority value of the requested interrupt is higher than the global interrupt threshold value. Therefore, Hobbs teaches taking into account a comparison of a priority of an interrupt and a priority of a currently active thread, whereas independent claim 1 takes into account a comparison of a unique interrupt priority value and a global interrupt threshold value.

Hobbs is therefore different from independent claim 1 in two respects. First, comparing an interrupt priority with thread priority, versus comparing an interrupt priority value with an interrupt threshold value. Second, Hobbs makes its comparison on an individual scale against each current thread, and therefore Hobbs teaches away from a comparison with a global threshold.

Further, Hobbs is directed to a different technique than the subject matter of independent claim 1. Hobbs teaches a multi threading software system, that is the threads are threads of a software program. On the other hand, independent claim 1 recites execution logic, that is a multi threading hardware system where the threads are processed by hardware logic.

Moreover, there would have been no motivation to combine Hobbs and Wilson. Hobbs and Wilson describe such completely different technical systems, it is difficult to imagine how a person skilled in the art would have considered combining the two references. Nevertheless, the Examiner argues Wilson, in column 2 line 65, to column 3, line 55, explicitly teaches specifying a global interrupt threshold value applicable to all threads. However, what Wilson teaches is providing a latency manager external of the processor between the processor and the memory subsystems. (See Wilson, column 3, lines 1-2.) The latency manager determines a time it would take to acquire a piece of data of a first process from the memory system. In other words, Wilson's comparison is in effect a comparison of whether the piece of data which is to be acquired is stored in level 1 or level 2 memory to determine the access time required to access that particular piece of data, with a threshold. Wilson therefore teaches only comparing pre-calculated latencies values for accessing a memory system with a threshold. Wilson does not disclose or suggest a global interrupt threshold value that is applicable to all of the plurality of active threads, and using the global interrupt threshold value to compare whether the priority value of a requested interrupt is higher than this global interrupt threshold value.

Even were a person skilled in the art to have combined Wilson and Hobbs, the result would have been the latency manager of Wilson determining the access times in the system of Hobbs. This skilled person certainly would not have modified the individual scheme of comparing interrupt threshold with a thread priority as outlined by Hobbs merely based on the teachings of Wilson.

Independent claim 1 is therefore patentable over Hobbs in view of Wilson for at least these reasons.

Since independent claims 22, 25, and 29 include limitations similar to the limitations discussed above with respect to independent claim 1, they are patentable over Hobbs in view of Wilson for at least the same reasons.

Claims 2-5, 9-16, 20, 21, 26, 30-32, 34, 35, and 37-40 depend from the independent claims, and are therefore patentable over Hobbs in view of Wilson for at least the same reasons.

Dependent claims 6-8, 17-19, 27, 28, 33, and 36, which depend on the independent claims, stand rejected under 35 U.S.C. § 103(a) in view of the additional Radhakrishna reference. Radhakrishna is not cited to cure the deficiencies in the combination of Hobbs and Wilson, but rather for its disclosure of other features, which, whether or not it does disclose, fails to cure the noted deficiencies in the Hobbs and Wilson combination. These dependent claims are therefore patentable over the applied references at least by virtue of their respective dependencies on the independent claims.

In view of the above, Applicant believes the pending application is in condition for allowance.

In the event a fee is required or if any additional fee during the prosecution of this application is not paid, the Patent Office is authorized to charge the underpayment to Deposit Account No. 50-2215.

Dated: January 8, 2010

Respectfully submitted,

By Laura C. Brutman

Laura C. Brutman
Registration No.: 38,395
DICKSTEIN SHAPIRO LLP
1633 Broadway
New York, New York 10019-6708
(212) 277-6500
Attorney for Applicant